Making Gnutella-like P2P Systems Scalable

Y. Chawathe, S. Ratnasamy,
L. Breslau, N. Lanham, S. Shenker

Presented by:
Herman Li
Mar 2, 2005

Outline

- What are peer-to-peer (P2P) systems?
- Early P2P systems
- Distributed Hash Table (DHT)
- GIA: Design & Implementation
- Simulation results
- Implementation results
- Conclusions
- Discussions

What are P2P systems?

- Distributed systems that have no central servers
- Peers in the system have equal functionalities and responsibilities
- Peers form an overlay network at the edge of the physical network
- Advantages: organic-scaling, reduce costly infrastructure, enable resource aggregation
- Examples of early P2P systems: Napster, Gnutella, KaZaA…

Early P2P systems

- Napster
  - Centralized file index; p2p file transfer
- Gnutella
  - Distributed file index; uses simple flooding
  - More download users than upload users
- KaZaA
  - Headquartered in Tuvalu, Australia
  - Supernodes and ordinary nodes; flood between supernodes

Distributed Hash Table (DHT)

- Hash table over the Internet
- Lookup requires $O(\log n)$ steps
- Peers join and leave network at will
- Examples: Chord, CAN, Pastry, Tapestry…

- Why not use DHT?
  - High churn rate
  - Need keyword searches
  - Look for popular objects

GIA (Gianduia)

- Goals
  - Scalable in size
  - Provides higher aggregate query rates
- Main differences
  - Nodes are close to high capacity nodes
  - Flow control scheme balances load
  - One-hop replication of content info
  - Biased random walks

Topology Adaptation

- Bootstrapping via host cache or equivalent schemes
- Constructs overlay network with low capability nodes close to high capability nodes
- Uses satisfaction metric: neighbours must have outgoing capacity to handle forwarded queries
  - Metric controls update interval

Flow Control

- Active flow control assigns tokens to neighbours
- Token allocation rate varies on query-processing capability and buffer queue
- Start-time Fair Queuing is used with weight as the neighbours' advertised capacity

One-hop Replication

- Content information is exchanged during connection, and updated incrementally
- High capacity peers can act as a proxy for low capacity peers

Search Protocol

- Random walk to highest capability peer with flow token received
- Uses GUID to send queries to different paths
- TTL and max_responses bounds propagation
- Advantage: Reduce flooding / congestion
- Disadvantage: Sensitive to peer failures
  - Solution: keep-alive messages, with app-level retries

Simulation Results

- Four models:
  - FLOOD: Simple flooding - Gnutella
  - RWRT: Random walk over random topologies
  - SUPER: Addition of supernodes – KaZaA
  - GIA: Proposed system

Node capacity distributions

Capacity level	1x	10x	100x	1000x	10000x
% of nodes	20%	45%	30%	4.9%	0.1%

Simulation Results

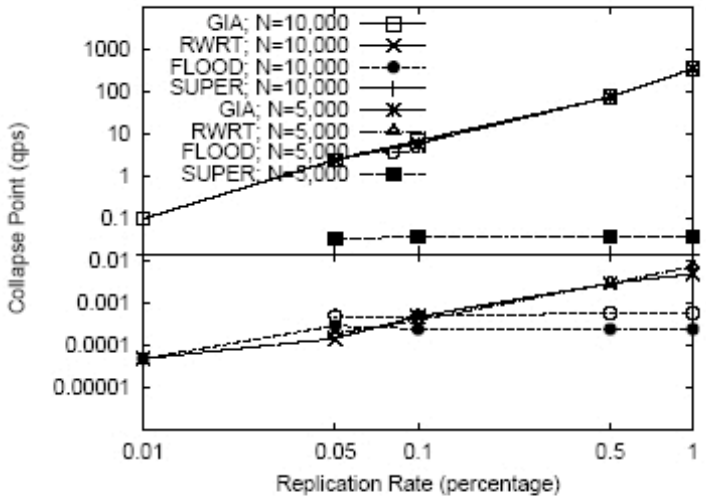

Figure 3: Comparison of collapse point for the different algorithms at varying replication rates and different system sizes.

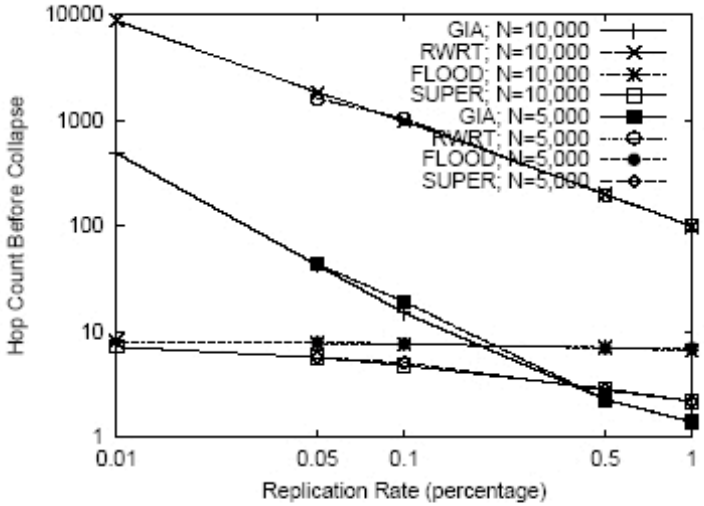

Figure 4: Hop-count before collapse.

Simulation Results

Repl. factor	MAX_ RESP.	GIA CP (CP-HC)	RWRT CP (CP-HC)	FLOOD CP	SUPER CP
1%	1	350 (1.4)	0.005 (98.7)	0.00025	0.015
1%	10	8 (12.5)	0.0004 (1020)	0.00025	0.015
1%	20	2.5 (28)	0.00015 (2157)	0.00025	0.015

Table 2: CP decreases with increasing numbers of requested answers (MAX_RESPONSES). The corresponding hop-counts before collapse for each case are shown in parentheses. Since hop-counts are ambiguous for FLOOD and SUPER when there are multiple responses, we ignore CP-HC for those cases.

Repl. factor	MAX_ RESPONSES	GIA CP	RWRT CP	FLOOD CP	SUPER CP
1%	10	8	0.0004	0.00025	0.015
0.1%	1	7	0.0005	0.00025	0.015
1%	20	2.5	0.00015	0.00025	0.015
0.05%	1	2.5	0.00015	0.00025	0.015

Table 3: A search for k responses at r% replication is equivalent to one for a single answer at $\frac{r}{k}$% replication.

Simulation Results

- Performance due to combination of schemes
- Biased random walk helps finding high capacity nodes in low query loads; flow control helps when query loads are high

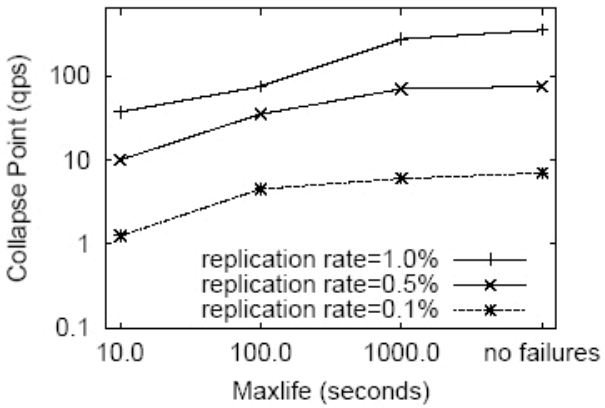

Figure 5: Collapse Point under increasing MAXLIFETIME **for a 10,000 node GIA system**

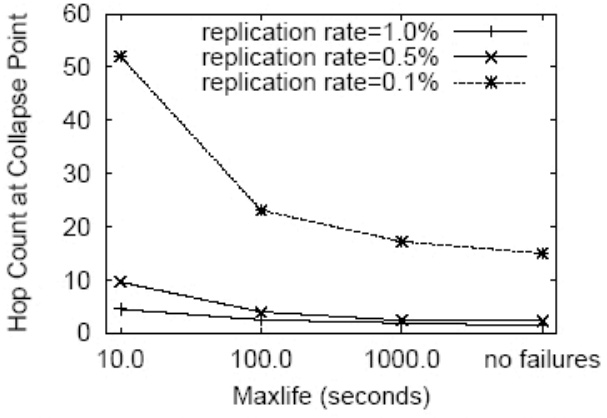

Figure 6: Hop Count under increasing MAXLIFETIME **for a 10,000 node GIA system**

Implementation Results

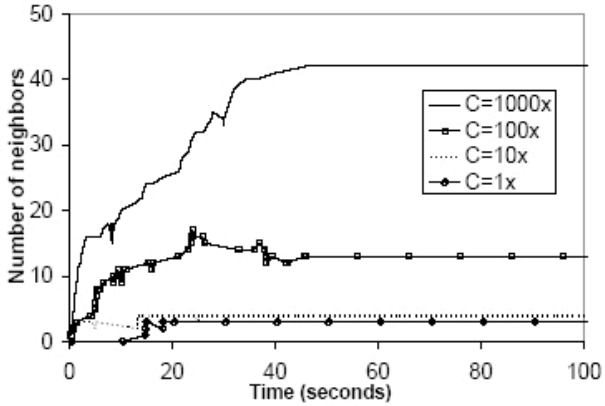

Figure 8: Progress of topology adaptation for an 83-node topology over time. The graph shows changes in the number of neighbors of four nodes (each with different capacities).

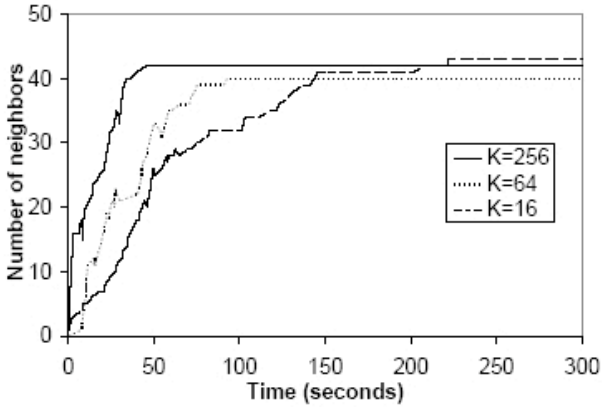

Figure 9: Progress of topology adaptation for a 1000x capacity node over time. The graph shows changes in the number of neighbors of a node over different runs of the experiment, each with a different value of K for the adaptation interval function.

Gnutella 0.6

- Leaf mode and ultrapeer mode
  - Leaf nodes connects to few ultrapeers
  - Ultrapeers connects to one another
  - Overlay network becomes smaller
  - Traffic to leaf nodes decreased

- “Exactly how to interpret the Search Criteria is not specified…”

Conclusions

- Additions: flow control, dynamic topology adaptation, one-hop replication and biased random walk
- 3 to 5 orders of magnitude improvement in total capacity
- Unstructured P2P systems are more suitable for mass-market file sharing

References

- Peer-to-Peer Computing, Dejan S. Milojicic et al.
- Scaling Unstructured Peer-to-Peer Networks With Multi-Tier Capacity-Aware Overlay, Mudhakar Srivatsa et al.
- Should we build Gnutella on a structured overlay? Miguel Castro et al.
- http://rfc-gnutella.sourceforge.net

Comments

- Pluses
  - Provides both simulation & implementation
  - Shown significant improvements
- Minuses
  - Latency not well studied
  - No distinction of logical distance vs physical distance
  - System does not scale well beyond knee point
  - Capability ignores CPU & I/O
  - Minimal experimental results on actual implementation
  - Many magic numbers used

Discussions

- Gnutella and GIA does not address the problem of potential partitioned networks. Can it be improved?
- Performance evaluation assumes 40-60 neighbours for each supernode. Is this reasonable?
- The random walk protocol is ad hoc. Is delay a necessary trade-off?
- Will today's P2P overlay network become tomorrow's mainstream network?
- Other points?